

OBSTRUCTIONS TO GENERICITY IN STUDY OF PARAMETRIC PROBLEMS IN CONTROL THEORY

VIKTOR LEVANDOVSKYY AND EVA ZERZ

CONTENTS

1. Introduction	1
2. Genericity of Gröbner Bases of Parametric Modules	3
2.1. LIFT and LEFTINVERSE Algorithms	4
2.2. Stratification of Obstructions to Genericity	6
2.3. The GENERICITY Algorithm	8
2.4. Comparison with Other Methods	9
3. Implementation of Algorithms	10
4. Example: Two Pendula, Mounted on a Cart	14
4.1. Negligible Friction	15
4.2. Essential Friction	15
5. Conclusion and Future Work	19
References	20

Dedicated to Michael S. Yakir

1. INTRODUCTION

Motivation: Problems from control theory often involve a set of physical parameters, for instance, masses, spring constants, and damping coefficients with mechanical systems, or resistances, capacitances, and inductances with electrical circuits. The structural properties of the control system may depend crucially on the specific choice of concrete parameter values. In many relevant examples, a system is generically controllable (i.e., controllable for almost all possible parameter values), but becomes uncontrollable when certain relations between the parameters are fulfilled.

It has been shown for many system classes of practical interest that controllability amounts to the torsion-freeness of a module associated to the system. For example, if $A = \mathbb{K}[\partial_1, \dots, \partial_n]$ for a field \mathbb{K} and $\mathcal{F} = \mathcal{C}^\infty(\mathbb{R}^n, \mathbb{K})$, the system given by the linear constant-coefficient partial differential equations $R(\partial_1, \dots, \partial_n)w = 0$, where $R \in A^{p \times q}$ and $w \in \mathcal{F}^q$, is controllable if and only if $M = A^{1 \times q} / A^{1 \times p} R$ is torsion-free. Then the **parametric controllability** problem can be formulated as follows: Given $R \in A^{p \times q}$, where $A = \mathbb{K}(p_1, \dots, p_t)[\partial_1, \dots, \partial_n]$ for some parameters p_1, \dots, p_t , find out whether M is generically torsion-free, and moreover, determine

Key words and phrases. generic properties, systems with parameters, Gröbner bases, parametric Gröbner bases, non-commutative Gröbner bases.

the relations among the p_i that will cause torsion elements in M . In general, we pose this question for a left module M over a (non-commutative) algebra A . The antipode notion of controllability is autonomy, which happens for systems being torsion. For a corresponding system module M , this means that M is annihilated by a non-zero ideal in A .

To a system module M over a system algebra A one associates the **transposed module** $N = N(M)$, defined as follows. Let the left module M be presented by a matrix $R \in A^{p \times q}$, then $N(M)$ is a right module with the presentation matrix $R^T \in A^{q \times p}$.

Then, there is an alternative description of torsion-freeness (controllability) and torsion (autonomy) in the language of homological algebra. Namely,

M is torsion-free if and only if $\text{Ext}_A^1(N(M), A) = 0$, and

M is a torsion module if and only if $\text{Ext}_A^0(M, A) = \text{Hom}_A(M, A) = 0$.

For a survey of the correspondence between control systems and their system modules, see [9] and the references therein, in particular, the works of Oberst, Pommaret and Quadrat. A general approach to parametric modules including the case when the parameters are non-constant was introduced in [14, 15]. The authors showed that so-called trees of integrability conditions, depending on parameters of the system, determine the control-theoretic properties of the system. These trees result into systems of partial differential equations and nonlinear differential conditions.

The situation described above was our original motivation for studying parameter-dependent questions of homological algebra such as the specific problem outlined above. It turned out that apart from the concrete application area, it is a challenging task for computer algebra to investigate parametric modules, and in particular, to get a grip on the special values of parameters that cause a qualitative change of structural module properties. These questions reach far beyond the limited set of algebras that typically arise in control theory. Roughly speaking, the problem can be tackled from the computational point of view for virtually every algebra that is accessible to Gröbner basis techniques. The main idea is simple but effective: it consists in a careful monitoring of denominators of cofactors that appear during Gröbner basis computation. Thus in this article we continue with the investigations, started in the articles [9, 10].

Outline of the paper: In this paper, we give an algorithm for answering the following

Question from Control Theory. Given a linear system S , depending on a finite number of parameters. Determine the control-theoretic properties (such as the decomposition into a controllable and an autonomous part) of S for all the values of involved parameters.

Since there is, for certain system classes [2, 10, 13], a one-to-one correspondence between control-theoretic properties of a system S and the homological properties of an associated module $M(S)$, we can reformulate the question as follows:

Question for Computer Algebra. Given a finite presentation of a parametric module M over a (non-commutative) algebra A , determine the properties (e.g. homological) of M for all the values of involved parameters.

We present detailed solutions for the bipendulum equations (Example 3.1), and for the "two pendula mounted on a cart" problem, for both negligible friction (Section 4.1) and essential friction (Section 4.2). The latter problem, to the best of our knowledge, has not been yet solved completely in an explicit way. We also present and comment on several curious examples. Namely, we show the existence of a non-generic controllability in the generically autonomous system (Example 3.2), and present a system, where both controllability and autonomy properties appear only in the non-generic situation (Example 3.3). In the treatment of the case 3 of 4.2, we illustrate the ability of our method to treat nested obstructions, that is, investigating sub-obstructions of a given obstruction to genericity.

Preliminaries: Algebraically speaking, a **parameter** is a non-zero (and thus invertible) element of the ground field. In this article we deal with parameters which mutually commute with the elements of the algebra. In other words, the action of operators of the algebra on parameters is just the commutative multiplication.

In this article, we use the following definition of a property being generic.

Definition 1.1. Let $\mathcal{P}(p_1, \dots, p_n)$ be a polynomial expression in p_i over some domain \mathcal{D} , on which a measure μ exists. The identity $\mathcal{P} = 0$ holds **generically** in \mathcal{D} if $\mathcal{P}(\xi_1, \dots, \xi_n) = 0$, for almost all $(\xi_1, \dots, \xi_n) \in \mathcal{D}^n$.

In other words, $\mathcal{P} = 0$ holds in $\mathcal{D}^n \setminus E$, where $E \subset \mathcal{D}^n$ and $\mu(E) = 0$.

For instance, if $\mathcal{D} = \mathbb{C}$ and \mathcal{P} vanishes on the complement of a nonzero algebraic set $E \subset \mathcal{D}^n$, then $\mathcal{P} = 0$ holds generically in \mathcal{D}^n . The Krull dimension Kr. dim of the coordinate ring $\mathbb{K}[V]$ of a variety V can be used for defining a measure on closed subsets $E \subset \mathcal{D}^n$ by assigning $\mu(E) = 0$ if $\text{Kr. dim } \mathbb{K}[E] < n$, and $\mu(E) = 1$ otherwise.

Notations: For a matrix M , M^T denotes its transposed matrix. By ${}_A\langle F \rangle$ we denote a left A -submodule, generated by the finite set F . The subscript A is dropped when A is commutative. For an ideal I in a commutative ring $\mathbb{K}[x_1, \dots, x_n]$, we denote by $V(I) \subseteq \mathbb{K}^n$ the set of common zeros of polynomials in I .

2. GENERICITY OF GRÖBNER BASES OF PARAMETRIC MODULES

Since the major role in computations (of e.g. homological properties) is played by Gröbner bases, we investigate their behaviour in the case when a ground field involves parameters.

Let \mathbb{K} be a field. Let A be a (non-commutative) algebra over $\mathbb{K}(p_1, \dots, p_t)$. Suppose that in this algebra the notion of algorithmic left Gröbner basis exists (e.g. A can be a ring of solvable type [6] or, more restrictively, an Ore algebra [2]).

Let us recall the definition of a ring of solvable type.

Definition 2.1. Let $K \supseteq \mathbb{K}$ be a skew field and let $R' := K[x_1, \dots, x_n]$ be a commutative ring over K . Suppose that \prec is a fixed term ordering on R' . Let R be a ring generated by $\{x_1, \dots, x_n\}$ subject to the new multiplication $*$. If the

properties 1 and 2 below hold and $(R, *)$ is an associative ring, R is called a **ring of solvable type**.

1. $\forall 1 \leq i < j \leq n$, $x_i * x_j = x_i x_j$ and $x_j * x_i = c_{ij} x_i x_j + p_{ij}$, where $0 \neq c_{ij} \in K$ and $p_{ij} \in R'$, such that $\text{lm}(p_{ij}) \prec x_i x_j$,
2. $\forall 1 \leq i \leq n$, $\forall a \in K$, $a * x_i = a x_i$ and $x_i * a = c_{ai} a x_i + p_{ai}$, where $0 \neq c_{ai} \in K$ and $p_{ai} \in K$.

Good examples of rings of solvable type are the rings of (partial) differential-difference operators.

The elements of a free module A^m are represented as the vectors $\bar{t} = \sum_{k=1}^m t_k e_k$, where $t_i \in A$, and e_i is the i -th canonical basis vector. By $\bar{0}$ we denote the zero vector $(0, \dots, 0)^T \in A^m$. The set of vectors $\bar{t}_1, \dots, \bar{t}_l$, for instance the set of generators of a submodule of a free module A^m , will be often identified with the matrix $T \subset \text{Mat}(m \times l, A)$. A single vector \bar{t}_i corresponds to the i -th column of T and vice versa.

Given a monomial well-ordering \prec on A , there are several ways to extend it to a monomial module ordering \prec_M on A^m , that is, an ordering consisting of two components (\prec, \prec_C) , where \prec_C is an ordering on the components e_i . In the following, we need a so-called **term over position ordering**, that is, $m_1 e_i \prec_M m_2 e_j$ if and only if $m_1 \prec m_2$ or, if $m_1 = m_2$, then $e_i \prec_C e_j$ for monomials $m_i \in A$.

Recall, that a **left syzygy** of a finite set of elements $\{f_1, \dots, f_m\}$, $f_i \in A$, is a tuple $(b_1, \dots, b_m)^T \in A^m$, such that $b_1 f_1 + \dots + b_m f_m = 0$. The set of all left syzygies of a given set of m elements is a left submodule of A^m . It is often denoted as $\text{Syz}(\{f_1, \dots, f_m\})$.

In this article, we work with left submodules, left syzygies etc. It is clearly possible to do the same also from the right. However, two-sided (bimodule) problems deserve, except for the commutative case, a fairly distinct treatment. Most (if not all) problems, originating from applications of e.g. control theory, are formulated in terms of left modules.

2.1. Lift and LeftInverse Algorithms.

Proposition 2.2. Suppose that a left submodule L of a free module A^m is generated by the set of column vectors $F = \{\bar{f}_1, \dots, \bar{f}_l\} \subset A^m$. Consider the set $\tilde{F} := \{\bar{f}_1 + e_{m+1}, \dots, \bar{f}_l + e_{m+l}\}$ and assume, that the fixed ordering \prec on A^m , naturally extended to the ordering \prec_l on A^{m+l} , satisfies $x^\alpha e_{m+i} \prec_l x^\beta e_j$, for all $1 \leq i \leq l$, $1 \leq j \leq m$ and for all α, β . Suppose that the left Gröbner basis \tilde{G} of \tilde{F} is finite. Then we reorder the columns of \tilde{G} in such a way, that the elements, whose first m components are zero, are moved to the left. This process is schematically presented in the following picture:

$$\tilde{F} = \begin{pmatrix} \bar{f}_1 & \dots & \bar{f}_l \\ 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \xrightarrow{\text{LEFTGB}} \left(\begin{array}{ccc|ccc} \bar{0} & \dots & \bar{0} & \bar{h}_1 & \dots & \bar{h}_t \\ \hline & \mathbf{S} & & & \mathbf{T} & \end{array} \right) = \tilde{G}.$$

Let $H = \{\bar{h}_1, \dots, \bar{h}_t\}$ be a left Gröbner basis of F . Recall that we identify F with the matrix $(\bar{f}_1, \dots, \bar{f}_l) \in A^{m \times l}$, T with an $l \times t$ matrix over A , and H with the matrix $(\bar{h}_1, \dots, \bar{h}_t) \in A^{m \times t}$. Then

- \mathbf{T} is a left transformation matrix between two generating sets of F , that is $H^T = \mathbf{T}^T F^T$ holds,
- the columns of \mathbf{S} form a left Gröbner basis of $\text{Syz}(\{\bar{f}_1, \dots, \bar{f}_l\})$.

Proof.

$$\text{Since } \bar{h}_i = \sum_{k=1}^l a_{ik} \bar{f}_k, \text{ we have } \sum_{k=1}^l a_{ik} (\bar{f}_k + e_{m+k}) = \bar{h}_i + \sum_{k=1}^l a_{ik} e_{m+k}.$$

Hence, the i -th column of \mathbf{T} is $(a_{i1}, \dots, a_{il})^T$, and $H^T = (\bar{h}_1, \dots, \bar{h}_t)^T = \mathbf{T}^T \cdot F^T$.

Let $\mathbf{S} = \{\bar{s}_1, \dots, \bar{s}_r\}$ and $\tilde{S} := \{(\bar{0}, \bar{s}_1)^T, \dots, (\bar{0}, \bar{s}_r)^T\}$. Since \tilde{S} is a left Gröbner basis of \tilde{F} , for any f in ${}_A\langle \tilde{F} \rangle \cap (\{0\}^m \times A^l) = {}_A\langle \tilde{F} \rangle \cap \oplus_{k=m+1}^{m+l} A e_k$ there exists $g \in \tilde{G}$, such that $\text{lm}(g)$ divides $\text{lm}(f)$. Then $\text{lm}(g) \in \{0\}^m \times A^l$, hence, by the property of the ordering, $g \in \tilde{G} \cap (\{0\} \times A^l) = \tilde{S}$. Thus \tilde{S} is a left Gröbner basis of $\tilde{F} \cap (\{0\}^m \times A^l)$ and, in particular, \tilde{S} generates the latter. Since

$$\sum_{k=1}^l b_k (\bar{f}_k + e_{m+k}) = \sum_{k=1}^l b_k e_{m+k} \text{ holds if and only if } \sum_{k=1}^l b_k \bar{f}_k = \bar{0},$$

\mathbf{S} consists of columns $(b_1, \dots, b_l)^T$, which are the syzygies of the set $\{\bar{f}_1, \dots, \bar{f}_l\}$. □

Remark 2.3. Clearly, for a Noetherian algebra A the algorithm terminates.

More generally, if the left Gröbner basis of F is finite, we get the transformation matrix in finitely many steps. Namely, in the generalized Buchberger's algorithm for computing left Gröbner basis, we do not consider S -polynomials between elements whose leading monomials include components greater than m .

If the algebra A is commutative, the transformation matrix property translates into $H = F \cdot \mathbf{T}$.

We call the algorithm computing the transformation matrix as above $\text{LIFT}(F, H)$. Note that with this algorithm we are able to trace any computation which uses Gröbner bases. It is worth mentioning that Proposition 2.2 shows, that with basically one Gröbner basis computation we can get three important objects, namely a Gröbner basis of a module, a Gröbner basis of the first syzygy module and a transformation matrix. These three applications are sometimes called **Gröbner trinity** and play a fundamental role in computer algebra.

Many problems in control theory involve parameters, which are known to be non-zero, or even strictly positive, for physical reasons. However, it might happen that the vanishing of certain algebraic expressions in the parameters has a direct impact on the control-theoretic properties. Very often we observe generically controllable parametric systems which, for some values of parameters, become uncontrollable.

As a further application of the algorithm LIFT , we compute a left inverse of a given polynomial matrix in the case it exists. Below, the algorithm $\text{RMLEFT-GROEBNERBASIS}(M)$ computes the monic reduced minimal left Gröbner basis of a submodule M , which is unique for a fixed ordering ([6, 7]).

Algorithm 2.4. LEFTINVERSE(matrix M)

Input: $M \in \text{Mat}_{m \times n}(A)$

Output: $L \in \text{Mat}_{n \times m}(A)$, such that $L \cdot M = Id_{n \times n}$
or $0 \in \text{Mat}_{1 \times 1}(A)$, if no left inverse exists

module $G := \text{RMLEFTGROEBNERBASIS}(M)$

if $G \neq Id_{n \times n}$ **then**

report "No left inverse exists"

return 0

endif

module $N := \text{TRANPOSE}(M)$

module $K := \text{LIFT}(N, Id_{n \times n})$

return $\text{TRANPOSE}(K)$

Proof. The algorithm LEFTINVERSE terminates as soon as LIFT does. Note that $LM = Id_{n \times n}$ can happen only in the case when the monic reduced minimal left Gröbner basis of a free submodule generated by the columns $\{N_j\}$ of $N = M^T$ is equal to $Id_{n \times n}$.

In the setup of the LIFT algorithm, we use $H = Id_{n \times n}$. Denote by K the result of $\text{LIFT}(N, Id_{n \times n})$. Then, by the Proposition 2.2, $Id_{n \times n} = Id_{n \times n}^T = K^T M^T = K^T N$. Hence, for $L = K^T$ we have $LM = Id_{n \times n}$. \square

The existence of a left inverse (or, more generally, a generalized inverse G , such that $G \cdot M \cdot G = G$) often gives us the information on genericity of parameters. Namely, one analyzes the possible vanishing of denominators of a generalized inverse, as it is done in e.g. [2]. In the special case where $A = \mathbb{K}[\partial]$ is a principal ideal domain, consider the module $M = A^{1 \times q} / A^{1 \times p} R$. Without loss of generality, we can assume R has full row rank. Then M is torsion-free if and only if there exists a right inverse to R .

As we have shown, computing the inverse is a special case of computing the transformation matrix with the algorithm LIFT. In comparison with LEFTINVERSE, LIFT allows us to deal effectively with more general problems.

We call the polynomials in parameters, whose vanishing implies the failure of generic properties, **obstructions to genericity**. We can compute them as described above using the LIFT algorithm.

There is a need for complete information on the parametric module. It consists of the list of properties, computed for the generic and all the non-generic cases. In the context of generically controllable problems, we are interested in computing e.g. an annihilator of a torsion submodule for the each non-generic case. Thus, we need to stratify the set of obstructions.

2.2. Stratification of Obstructions to Genericity. Let \mathbb{K} be a field of characteristic 0. Recall that a set is called **locally closed**, if it is a difference of two closed sets. A finite union of locally closed sets is called a **constructible** set.

Suppose we are given a set of polynomials $P = \{p_1, \dots, p_n\} \subset \mathbb{K}[a_1, \dots, a_m]$, which are irreducible over \mathbb{K} .

We associate to P a set $C(P) := \{\bar{\xi} = (\xi_1, \dots, \xi_m) \in \mathbb{K}^m \mid \prod_{i=1}^n p_i(\bar{\xi}) = 0\}$.

Lemma 2.5. The set $C(P)$ is constructible.

Proof. Let $\Omega := \{(\Lambda', \Lambda'') \mid \Lambda' \cup \Lambda'' = \{1, \dots, n\}, \Lambda' \cap \Lambda'' = \emptyset\}$ be the set of all divisions of $\{1, \dots, n\}$ into two disjoint complementary subsets. Let, furthermore, $\Sigma := \Omega \setminus (\emptyset, \{1, \dots, n\})$. Then,

$$\begin{aligned} C(P) &= \bigcup_{(j,k) \in \Sigma} \{\bar{\xi} \mid \forall j \in \Lambda' \ p_j(\bar{\xi}) = 0, \quad \forall k \in \Lambda'', \ p_k(\bar{\xi}) \neq 0\} = \\ &= \bigcup_{(j,k) \in \Sigma} V(\langle \{p_j \mid j \in \Lambda'\} \rangle) \setminus V(\langle \{p_k \mid k \in \Lambda''\} \rangle) = \\ &= \bigcup_{(j,k) \in \Sigma} \cap_{j \in \Lambda'} V(\langle p_j \rangle) \setminus \cap_{k \in \Lambda''} V(\langle p_k \rangle), \end{aligned}$$

and, indeed, we see that $C(P)$ is a disjoint union of locally closed sets. Note that in $C(P)$ there is a closed subset $\cap_i V(p_i)$; the rest of subsets are locally closed. \square

It is convenient to represent $C(P)$ as a binary tree, where the vertices are the decision points, associated to polynomials p_i , and the edges represent the logical conditions ($p_i = 0$) and ($p_i \neq 0$), respectively. In such a way it is easy to see, that starting from n elements in the set P , we will have $2^n - 1$ algebraic systems describing the locally closed components of $C(P)$.

Given two ideals $I, J \in \mathbb{K}[a_1, \dots, a_m]$, an algebraic data describing a locally closed set $V(I) \setminus V(J) = V(I) \setminus (V(I) \cap V(J))$ can be computed with a factorizing Gröbner basis algorithm (e.g. [5]). Such an algorithm takes I, J as input and returns a list of ideals, where the zero set of the intersection of the output ideals is contained in the $V(I)$ and contains the complement of the $V(J)$ in $V(I)$. We refer to this algorithm as to $\text{FACTGB}(I, J)$.

Example 2.6. Let $P = \{p_1, p_2\}$, then the binary tree for $C(P)$ consists of the following 3 systems of equations and inequations: $\{p_1 = 0, p_2 = 0\}$, $\{p_1 \neq 0, p_2 = 0\}$ and $\{p_1 = 0, p_2 \neq 0\}$.

Denote $V_i := V(\langle p_i \rangle)$ for $i = 1, 2$, and $V_{12} := V(\langle p_1, p_2 \rangle) = V_1 \cap V_2$. Then, the decomposition of $C(P)$ can be written as $V_{12} \uplus (V_1 \setminus V_2) \uplus (V_2 \setminus V_1) = V_{12} \uplus (V_1 \setminus V_{12}) \uplus (V_2 \setminus V_{12})$, where \uplus denotes the disjoint union.

Computationally, we need to compute the Gröbner basis of an ideal $I_{12} := \langle p_1, p_2 \rangle$, and two lists $L_i := \text{FACTGB}(\{p_i\}, I_{12})$, obtained with the factorizing Gröbner basis algorithm, which describe $V_i \setminus V_{12}$.

Given a set of polynomials $\{f_1, \dots, f_s\} \subset \mathbb{K}[a_1, \dots, a_m]$, we factorize them and form a set of pairwise different irreducible factors $P := \{p_1, \dots, p_n\}$. We sort p_i by using a positively graded degree ordering, starting with the smaller elements. With such an ordering, it is easier to compute with locally closed sets. Namely, the bigger elements will often reduce to simpler polynomials with respect to the smaller elements. Thus, also the detection of empty components (that is, systems with no solutions) can be achieved faster.

Lemma 2.5 is constructive indeed. Together with the presentation of locally closed sets using the algorithm FACTGB above, we call the whole procedure $\text{STRATIFYLC}(\text{list } L)$. It takes a finite list of irreducible polynomials on the input and returns a list of systems of equations and inequations, corresponding to $C(P)$.

2.3. The Genericity Algorithm. Let A be a \mathbb{K} -algebra and suppose that the coefficients of a given system S involve parameters p_1, \dots, p_t . We interpret the parameters as generators of the transcendental field extension of \mathbb{K} and we use the natural $\mathbb{K}(p_1, \dots, p_t)$ -algebra structure on A .

Algorithm 2.7. GENERICITY(matrix M)

Assume, that a monomial module ordering on the algebra A is fixed.

Input: $M \in \text{Mat}_{m \times n}(A)$

Output: $\{h_1, \dots, h_s\} \subset \mathbb{K}[p_1, \dots, p_t]$, such that if a specialization of the parameters implies $h_i(p_1, \dots, p_t) = 0$, then a left Gröbner basis of M is different from the generic one

```

module  $G := \text{RMLEFTGROEBNERBASIS}(M)$        $\backslash\backslash G = \{\bar{g}_1, \dots, \bar{g}_\ell\} \in \text{Mat}_{m \times \ell}(A)$ 
matrix  $T := \text{LIFT}(M, G)$                    $\backslash\backslash T \in \text{Mat}_{n \times \ell}(A)$ 
list  $S, H$ ;  int  $i, j$ 
for  $j = 1$  to  $\ell$ 
     $i :=$  the leading component of  $\bar{g}_j$ 
    for  $k = 1$  to  $n$ 
        if  $(M_{ik} \neq 0)$  then
            if  $(T_{kj} \neq 0)$  then  $S := S \cup \text{DENOMINATOR}(T_{kj})$ 
            end if
        end if
    end for
end for
if  $(S \neq \text{empty list})$  then
     $H := \text{FACTORIZE}(S)$ 
     $H := \text{SIMPLIFY}(H)$ 
else  $H := \text{empty list}$ 
return  $H$ 

```

Proof. The algorithm $\text{FACTORIZE}(\text{list } L)$ returns a list of monic factors of every polynomial of the list L . The algorithm $\text{SIMPLIFY}(\text{list } L)$ refines a list L by removing doubled appearances of same elements. We may assume it also sorts L by an ordering, putting with the smaller elements in the beginning of the output.

The algorithm GENERICITY terminates as soon as LIFT terminates. Now, we prove the correctness. Suppose that the leading term of \bar{g}_j lies in the i -th module component. From the property $G^T = T^T M^T$ it follows, that there is a presentation of the element $G_{ij} \in A$ as the sum

$$G_{ij} = \sum_{k=1}^n T_{kj} M_{ik} = \sum_{k=1}^n T_{jk}^T M_{ki}^T.$$

Hence, it suffices to collect only the denominators of $T_{kj} \neq 0$ with $M_{ik} \neq 0$, since only such elements contribute to the leading coefficient of \bar{g}_j .

If some leading coefficient of the unique generic Gröbner basis vanishes for some specialization of parameters, then the Gröbner basis under such a specialization is different from the generic one. \square

Note that with the algorithm we obtain the expressions in the parameters which lead to non-generic Gröbner bases. In order to obtain Gröbner bases under specialization, provided by h_i , one cannot use the generic Gröbner basis. Instead, one has to compute the specialized Gröbner basis from scratch.

Suppose that the output of `GENERICITY` is the list of irreducible polynomials H . In practice, we exclude from H the polynomials, which do not satisfy the problem-specific constraints for e.g. physical admissibility like non-negativity. Then, we apply the algorithm `STRATIFYLC(H)` and obtain a complete stratification of a given system with respect to its parameters.

2.4. Comparison with Other Methods.

2.4.1. Comprehensive Gröbner bases. Comprehensive Gröbner bases (see e.g. [18]) were introduced by Weispfenning and generalized to rings of solvable type by Kredel [6].

A comprehensive Gröbner basis, by definition, is a finite subset G of a parametric polynomial ideal I such that $\sigma(G)$ constitutes a Gröbner basis of the ideal generated by $\sigma(I)$ under all specializations σ of the parameters in arbitrary fields ([18]).

The construction of a comprehensive Gröbner basis follows the lines of Buchberger's algorithm. However, the result will be a union of trees of ideal bases (called Gröbner systems), where each basis is accompanied with a set of conditions of parameters. Being a powerful theoretical instrument, comprehensive Gröbner bases are quite complicated to compute. To the best of our knowledge, there is no implementation yet, which is able to treat serious examples.

In our approach we separate two processes, which are unified in the comprehensive Gröbner basis method. Namely, we compute the tree of sets of conditions of parameters *after* the Gröbner basis and transformation matrix computations. In such a way we avoid repeated computations in trees of ideals and sets of conditions, which might occur during the computation of a comprehensive Gröbner basis.

2.4.2. The Leykin–Walther Method. The method has been formalized by Leykin for the case of ideals [11] and has been generalized to modules by Walther [17]. The idea behind the method has been used before, however Leykin and Walther formulated and proved the whole framework in a complete way. In the following, we reformulate the Lemma 2.3 from [17].

Let \mathbb{K} be a field of characteristic 0. Given a \mathbb{K} -algebra A , we consider parameters as new commutative variables and perform further computations in the \mathbb{K} -algebra $\tilde{A} := A \otimes_{\mathbb{K}} \mathbb{K}[p_1, \dots, p_m]$. We use in \tilde{A} an elimination ordering \prec_A for the variables of A . Such an ordering is characterized by the property $p_1^{\alpha_1} \dots p_m^{\alpha_m} \prec_A t$, for any monomial $t \in A$ and any $\alpha \in \mathbb{N}^m$.

Let $G = \{g_1, \dots, g_\ell\}$ be a reduced Gröbner basis for the left submodule $N \subset \tilde{A}^s$ with respect to the position over term ordering, induced by \prec_A on \tilde{A}^s . Let, moreover, $Q_N \subset \mathbb{K}[p_1, \dots, p_t]$ be the ideal $\{p \in \mathbb{K}[p_1, \dots, p_t] \mid p\tilde{A}^s \subseteq N\}$. For $g_i \notin Q_N\tilde{A}^s$, multiply all the leading coefficients with respect to $\mathbb{K}[p_1, \dots, p_t]$ of such g_i and denote the result by h . Let $\sigma : \mathbb{K}[p_1, \dots, p_t] \rightarrow \mathbb{K}$ be a specialization, then if $\sigma(h) \neq 0$,

then $\sigma(G) = \{\sigma(g_1), \dots, \sigma(g_t)\}$ is a Gröbner basis.

This method has some drawbacks in practice. Suppose that the number of parameters is big and there are many obstructions, which appear in several components as, say, leading coefficients by a monomial 1. This situation is typical for generically controllable systems. Then, using the method of Leykin–Walther, we are forced to compute Gröbner basis of a submodule of elements as described above, whereas a better solution would be just to collect the leading coefficients in parameters. Secondly, in a similar situation we get many elements in Gröbner basis and the analysis of the impact of obstructions, e.g. the computation of the stratification, becomes very involved.

On the other hand, this method allows us to handle the cases, when the parameters satisfy algebraic identities between themselves or when there are more general identities, involving both variables and parameters. We believe, that this method will be enhanced in order to overcome the described difficulties.

3. IMPLEMENTATION OF ALGORITHMS

The described method for detecting the obstructions to genericity of parametric modules is implemented in the procedure **genericity** of control theory toolbox `CONTROL.LIB` [1], which is realized as a library in the computer algebra system `SINGULAR` [4]. `SINGULAR` is the specialized computer algebra system for polynomial computations, well-known for its high performance (especially in Gröbner bases-related computations) and rich functionality. It uses intuitive C-like programming language, in which the libraries are written. It is important to mention, that `SINGULAR` is distributed under GPL license, that is, it is free for academic purposes.

The current implementation of the procedure **genericity** works in a little different way, compared with the Algorithm 2.7. Namely, it takes as input a matrix T , which is assumed to be the result of the `LIFT` algorithm. This minor modification allows us to compute the data, which are independent from the choice of a monomial module ordering. The output of the procedure **genericity** is a list of strings and thus it is ring-independent. In the first item of the list the names of parameters, by which we have divided in the algorithm, are collected. Every further item of the list contains a single non-trivial polynomial in the parameters.

There are several algorithms in `SINGULAR`, which compute (left) Gröbner bases of modules over commutative polynomial algebras and non-commutative *GR*-algebras [7, 8]. It is recommended to use the heuristic routine **groebner**, which often provides the best match for a concrete example. For more details on `SINGULAR`, consult with the book [5] and with the website of the system [4], which contains among other the online documentation. The algorithm `FACTGB` is implemented in `SINGULAR` and is accessible via the function **facstd**.

In the library `CONTROL.LIB`, we have implemented several functions for supporting the research in systems and control theory. Among others, there are the procedures **LeftInverse** and **LeftKernel**, their counterparts **RightInverse** and **RightKernel**, as well as **canonize** and **iostruct**.

The main purpose of the library is to provide maximal relevant information based on a simple input. This principle led us to the development of heuristic procedures `control` and `autonom`, which use homological computations. Respectively, for systems with a full row rank presentation matrix, there are dimension-guided procedures `controlDim` and `autonomDim`.

Given a system algebra and a system module over it, both procedures compute relevant properties of a given module from the point of view of controllability (with the procedure `control` or `controlDim`) or autonomy analysis (with the procedure `autonom` or `autonomDim`). The procedure `canonize` takes the output of either `control` or `autonom` procedure and computes reduced and tail-reduced Gröbner bases of the objects, thus simplifying and *canonizing* the output.

We illustrate the functionality of the library and the flexibility of SINGULAR with the following example.

Example 3.1. Consider a bipendulum, that is, a system, describing a bar with two fixed pendula of length ℓ_1 and ℓ_2 respectively (e.g. [13, 3]). The system algebra is a commutative algebra in variable ∂ over a field of rationals with parameters g, ℓ_1, ℓ_2 , that is, $\mathbb{Q}(g, \ell_1, \ell_2)[\partial]$. A system module is presented via the matrix $\begin{pmatrix} \partial^2 + \frac{g}{\ell_1} & 0 & -\frac{g}{\ell_1} \\ 0 & \partial^2 + \frac{g}{\ell_2} & -\frac{g}{\ell_2} \end{pmatrix}$. We run the following code in a SINGULAR session.

```
LIB "control.lib";
option(redSB); option(redTail);
```

With the LIB command we load the library. The `option` commands tell SINGULAR to compute reduced bases (`option(redSB)`), and also reduce not only leading terms, but any terms in the occurring polynomials (`option(redTail)`).

It is important to mention, that any polynomial computation in SINGULAR requires the definition of a ground ring.

```
ring r1 = (0,g,11,12),(d),(c,dp);
module RR = [d^2+g/11, 0, -g/11], [0, d^2+g/12, -g/12];
```

The ring we set bears the name `r1`, it has $\mathbb{Q}(g, \ell_1, \ell_2)$ as the ground field (0 stands for the characteristic of a field, `g`, `11`, `12` is a list of names for parameters), and the only variable `d`. The last comma-separated block describes the monomial module ordering on `r1`. In this case `(c,dp)` means the following. We use the descending ordering `c` on the module components and the degree reverse lexicographical ordering `dp` on the monomials in the same component.

```
module R = transpose(RR);
list L = canonize(control(R));
L;
```

We have to transpose the module `R`, because SINGULAR takes the columns of a given matrix presentation as the generators of a module. Here is the output of SINGULAR:

```
[1]:
    number of first nonzero Ext:
[2]:
    -1
[3]:
    strongly controllable(flat), image representation:
[4]:
    _[1]=[(-g*12)*d^2+(-g^2),(-g*11)*d^2+(-g^2),
```

```

      (-11*12)*d^4+(-g*11-g*12)*d^2+(-g^2)]
[5]:
  left inverse to image representation:
[6]:
  _[1,1]=(-11)/(g^2*11-g^2*12)
  _[1,2]=(12)/(g^2*11-g^2*12)
  _[1,3]=0
[7]:
  dimension of the system:
[8]:
  1
[9]:
  Parameter constellations which might lead to a non-controllable system:
[10]:
  [1]:
    g
  [2]:
    11-12

```

As one can see, in the output of the procedure we provide both textual comments on the properties of a system and the corresponding data. The heuristics says that the modules $\text{Ext}_{r,1}^i(R, r1)$ of a transposed module indeed vanish for $i \geq 1$ (-1 is returned in this situation). Hence, the system is generically controllable (the notion of strong controllability from above coincides with classical controllability for systems of ordinary differential equations). Moreover, the procedure computes the image representation, left inverse to the image representation and the dimension of the system. The 10-th item is the output of the procedure **genericity**, that is, a list of strings. The polynomial obstruction to genericity in this example is $\ell_1 - \ell_2$. The monomial obstruction g is not physically admissible.

Let us analyze the properties of the system in the non-generic case $\ell_1 = \ell_2 = \ell$. We do this with the help of the following code in the same SINGULAR session:

```

ring r2 = (0,g,1),(d),(c,dp);
module RR = [d^2+g/1, 0, -g/1], [0, d^2+g/1, -g/1];
module R = transpose(RR);
list L = canonize(control(R));
L;

```

We get the following output:

```

[1]:
  number of first nonzero Ext:
[2]:
  1
[3]:
  not controllable , image representation for controllable part:
[4]:
  _[1]=[ (g), (g), (1)*d2+(g) ]
[5]:
  kernel representation for controllable part:
[6]:
  _[1]=[0,1]
  _[2]=[1]
[7]:

```

```

    obstruction to controllability
[8]:
    _[1]=[0,1]
    _[2]=[(-1)*d2+(-g)]
[9]:
    annihilator of torsion module (of obstruction to controllability)
[10]:
    _[1]=(-1)*d2+(-g)
[11]:
    dimension of the system:
[12]:
    1

```

We see that the system is not controllable, since it contains a torsion submodule annihilated by $\langle \ell\partial^2 + g \rangle$. However, we give both image and kernel representations for the controllable part of the system and describe the obstruction to controllability explicitly. Now, we are interested in the autonomy analysis of this non-controllable system, what can be achieved with the following code:

```

list A = canonize(autonom(R));
A;

```

This gives us the following output:

```

[1]:
    number of first nonzero Ext:
[2]:
    0
[3]:
    not autonomous
[4]:
    kernel representation for controllable part
[5]:
    _[1]=[0,1]
    _[2]=[(-1)*d2+(-g),-1]
    _[3]=[g]
[6]:
    column rank of the matrix
[7]:
    2
[8]:
    dimension of the system:
[9]:
    1

```

Since the 0-th Ext module of the system module RR (in other words, $\text{Hom}_{r_2}(RR, r_2)$) does not vanish, the system is not autonomous. In addition, we compute a kernel representation for the controllable part, the column rank of the presentation matrix and the dimension of the system.

Parametric systems quite often are generically controllable and contain an autonomous subsystem for some special values of parameters. In the following example, we show that also a generically autonomous system might be controllable in a non-generic case.

Example 3.2. Let $R = \mathbb{K}(a, b)[\partial]$ be a ring. A module $N = R/\langle a\partial + b \rangle$ is generically autonomous. However,

if $a = 0, b \neq 0$, then $M = 0$ and thus M is autonomous,

if $a = 0, b = 0$, then M is free of rank 1 and hence M is controllable.

A general system might specialize to controllable and autonomous system in non-generic cases, as the next example shows.

Example 3.3. Let $R = \mathbb{K}(a, b)[\partial]$ be a ring. Consider a module $M = R^2 / \begin{pmatrix} 0 & 0 \\ 0 & a\partial + b \end{pmatrix}$. Generically, it is neither controllable nor autonomous, the annihilator of a torsion submodule is $\langle a\partial + b \rangle$.

The stratification of M with respect to parameters looks as follows:

if $a \neq 0, b = 0$, a torsion submodule of M is annihilated by $\langle \partial \rangle$,

if $a = 0, b \neq 0$, then M is free of rank 1,

if $a = 0, b = 0$, then M is free of rank 2.

Assume, that $a, b \in \mathcal{D} \supseteq \mathbb{K}$. Then the space of parameters \mathcal{D}^2 decomposes into a direct sum of subspaces $G \uplus E_1 \uplus E_2 \uplus E_3$, where $G = \{(a, b) \mid a \neq 0, b \neq 0\}$, $E_1 = \{(a, b) \mid a = 0, b \neq 0\}$, $E_2 = \{(a, b) \mid a \neq 0, b = 0\}$ and $E_3 = \{(a, b) \mid a = 0, b = 0\}$. Denote by \bar{E}_i the closure of E_i , then $\dim \bar{E}_1 = \dim \bar{E}_2 = 1$, $\dim \bar{E}_3 = 0$ in \mathcal{D}^2 . Hence, all \bar{E}_i have measure 0 and $\bar{G} = \mathcal{D}^2$ has measure 1.

Remark 3.4. There are packages like D -modules for MACAULAY2, [16], and ORE-MODULES for MAPLE, [3], which have a functionality to treat some of the problems above. The latter package provides the possibility to reveal dangerous parametric denominators via the computation of generalized inverse.

4. EXAMPLE: TWO PENDULA, MOUNTED ON A CART

Consider the Example 5.2.28 from [12] (see also the examples and solutions to them in [3]) describing two pendula, mounted on a cart.

In this example, m_i is the mass and L_i is the length of the i -th pendula. Respectively, k_i and d_i are the coefficients, characterizing the friction at the joints of pendula. M_0 denotes the mass of the cart and g is a gravitational constant. All these parameters can take only non-negative values.

Let us denote $z_i := k_i - m_i L_i g$ for $i = 1, 2$. Then the presentation matrix for a system module is constituted by the rows of the following matrix

$$\begin{pmatrix} m_1 L_1 \partial^2 & m_2 L_2 \partial^2 & (m_1 + m_2 + M_0) \partial^2 & -1 \\ m_1 L_1^2 \partial^2 + d_1 \partial + z_1 & 0 & m_1 L_1 \partial^2 & 0 \\ 0 & m_2 L_2^2 \partial^2 + d_2 \partial + z_2 & m_2 L_2 \partial^2 & 0 \end{pmatrix}$$

We take the transposed module of the matrix. It is convenient to consider the columns of the matrix above as the generators of submodule of a free module. Since the last generator then is just $(-1, 0, 0)^T$, we perform reduction and simplification of first components with respect to this generator. In such a way we obtain much easier presentation matrix.

4.1. Negligible Friction. Let us assume, that the friction is negligible (that is, $d_i = 0$ and $k_i = 0$). We get the simplified presentation matrix of the transposed module as follows:

$$\begin{pmatrix} L_1\partial^2 - g & 0 & m_1L_1\partial^2 \\ 0 & L_2\partial^2 - g & m_2L_2\partial^2 \end{pmatrix}$$

The generic reduced minimal Gröbner basis is the 2×2 identity matrix. With the LIFT algorithm we obtain the transformation matrix

$$\begin{pmatrix} \frac{L_1L_2}{g^2L_1 - g^2L_2}\partial^2 - \frac{1}{g} & -\frac{m_1L_1L_2}{g^2m_2L_1 - g^2m_2L_2}\partial^2 \\ \frac{m_2L_1L_2}{g^2m_1L_1 - g^2m_1L_2}\partial^2 & -\frac{L_1L_2}{g^2L_1 - g^2L_2}\partial^2 - \frac{1}{g} \\ -\frac{L_1L_2}{g^2m_1(L_1 - L_2)}\partial^2 + \frac{L_1}{gm_1(L_1 - L_2)} & \frac{L_1L_2}{g^2m_2(L_1 - L_2)}\partial^2 - \frac{L_2}{gm_2(L_1 - L_2)} \end{pmatrix}$$

Collecting the denominators, we can see that their lcm is $m_1m_2g^2(L_1 - L_2)$. Since m_i and g are strictly positive, the only obstruction to genericity appears when $L_1 - L_2 = 0$.

Indeed, in the case $L_1 = L_2 = L$ the generic Gröbner basis is $\begin{pmatrix} 0 & 1 \\ L\partial^2 - g & \frac{m_2}{m_1} \end{pmatrix}$, hence the system is not controllable. The torsion submodule is annihilated by the ideal $\langle L\partial^2 - g \rangle$, but the system is not completely autonomous.

4.2. Essential Friction. Now, all the parameters are strictly positive. The simplified presentation matrix of the transposed module is the following

$$\begin{pmatrix} L_1\partial^2 + d'_1\partial + z'_1 & 0 & m_1L_1\partial^2 \\ 0 & L_2\partial^2 + d'_2\partial + z'_2 & m_2L_2\partial^2 \end{pmatrix}$$

where $z'_i := \frac{z_i}{m_iL_i} = \frac{k_i}{m_iL_i} - g$ and $d'_i := \frac{d_i}{m_iL_i}$ for $i = 1, 2$.

The generic reduced minimal Gröbner basis is the 2×2 identity matrix. The output of the Algorithm 2.7 delivers the list of three polynomials $\{z'_1, z'_2, P\}$, where

$$P = L_1^2z'^2_2 - 2L_1L_2z'_1z'_2 - L_1d'_1d'_2z'_2 + L_1d'^2_2z'_1 + L_2^2z'^2_1 + L_2d'^2_1z'_2 - L_2d'_1d'_2z'_1.$$

$z'_i = 0$ means, that $k_i = m_iL_i g$. This is physically admissible situation. Let us analyze P for the admissibility. Indeed, P is irreducible but it has a special form, namely

$$(1) \quad P = (L_2z'_1 - L_1z'_2)^2 + (L_2d'_1 - L_1d'_2) \cdot (d'_1z'_2 - d'_2z'_1).$$

In particular, P vanishes if both z'_1 and z'_2 do, so P is admissible.

The stratification consists of 6 cases, namely

- (1) $k_1 = m_1L_1g, k_2 = m_2L_2g, P = 0$
- (2) $k_1 = m_1L_1g, k_2 \neq m_2L_2g, P = 0$
- (3) $k_1 = m_1L_1g, k_2 \neq m_2L_2g, P \neq 0$
- (4) $k_1 \neq m_1L_1g, k_2 = m_2L_2g, P = 0$
- (5) $k_1 \neq m_1L_1g, k_2 = m_2L_2g, P \neq 0$

$$(6) \quad k_1 \neq m_1 L_1 g, k_2 \neq m_2 L_2 g, P = 0$$

The setup for SINGULAR treatment of the cases is the following:

```
LIB "control.lib";
ring T = (0,g),(m1,m2,L1,L2,d1,d2,k1,k2),dp;
poly P = k1^2*L2^4*m2^2-2*k1*k2*L1^2*L2^2*m1*m2-k1*d1*d2*L2^2*m2+
k1*d2^2*L1^2*m1+2*k1*g*L1^2*L2^3*m1*m2^2-2*k1*g*L1*L2^4*m1*m2^2+
k2^2*L1^4*m1^2+k2*d1^2*L2^2*m2-k2*d1*d2*L1^2*m1-2*k2*g*L1^4*L2*m1^2*m2+
2*k2*g*L1^3*L2^2*m1^2*m2-d1^2*g*L2^3*m2^2+d1*d2*g*L1^2*L2*m1*m2+
d1*d2*g*L1*L2^2*m1*m2-d2^2*g*L1^3*m1^2+g^2*L1^4*L2^2*m1^2*m2^2-
2*g^2*L1^3*L2^3*m1^2*m2^2+g^2*L1^2*L2^4*m1^2*m2^2;
poly z1 = k1 - m1*L1*g;
poly z2 = k2 - m2*L2*g;
```

In particular, we can see the expression for P in terms of original variables. The name of a ring, where the interesting parameters live as polynomials, is T . In SINGULAR, we can switch between different rings and also map objects.

Case 1). $k_1 = m_1 L_1 g, k_2 = m_2 L_2 g, P = 0$.

Note that these three equations describe an algebraic variety, that is a closed set. The Gröbner basis of the ideal $k_1 - m_1 L_1 g, k_2 - m_2 L_2 g, P$ is $k_1 - m_1 L_1 g, k_2 - m_2 L_2 g$, since P vanishes, when both $k_1 = m_1 L_1 g$ and $k_2 = m_2 L_2 g$. Hence, it suffices to plug the values for k_i in the corresponding system. For this, we run the following code:

```
ring r1 = (0,g,m1,m2,L1,L2,d1,d2,k1,k2),(d),(c,dp);
poly z1 = 0; poly z2 = 0;
module RR =
    [m1*L1^2*d^2+d1*d+z1, 0, m1*L1*d^2],
    [0, m2*L2^2*d^2+d2*d+z2, m2*L2*d^2];
module R = transpose(RR);
list LC = canonize(control(R));
list LA = canonize(autonom(R));
```

From the output of `control` and `autonom` procedures, we conclude, that this system is neither controllable nor autonomous. In particular, the torsion submodule is annihilated by $\langle \partial \rangle$.

Case 2). $k_1 = m_1 L_1 g, k_2 \neq m_2 L_2 g, P = 0$.

Here we deal with the locally closed set $V(\langle k_1 - m_1 L_1 g, P \rangle) \setminus V(\langle k_2 - m_2 L_2 g \rangle)$. Using the following code, we get its better description. We employ a technical trick by modifying a ground ring in such a way, that k_i have priority over the rest of polynomials. In such a way during the computations the relation $k_1 = m_1 L_1 g$ will be used as replacing k_1 with $m_1 L_1 g$. This is achieved by using a different ordering like e.g. the **elimination** ordering (see e.g. [5]) for k_1, k_2 .

```
ring T2 = (0,g),(k1,k2,m1,m2,L1,L2,d1,d2),(a(1,1),dp);
poly z1 = ...; poly z2 = ...; poly P = ...; // we copy them from above
ideal I2 = P,z1;
I2 = groebner(I2);
facstd(I2,z2);
```

The output of `facstd` command gives us the only component

```
[1]:
_ [1]=k1+(-g)*m1*L1
_ [2]=k2*m1^2*L1^4+(-g)*m1^2*m2*L1^4*L2+m2*L2^2*d1^2-m1*L1^2*d1*d2
```

We are able to extract e.g. k_2 from the last equation explicitly:

$$k_2 = m_2 L_2 g + \frac{m_1 L_1^2 d_2 - m_2 L_2^2 d_1}{m_1^2 L_1^4} d_1$$

Alternatively, we can express d_2 in terms of variables m_i, L_i, k_2, d_1 .

Computing with substitutions, we see that this system is neither controllable nor autonomous. The torsion submodule is annihilated by $\langle m_1 L_1^2 \partial^2 + d_1 \partial \rangle$.

Case 3). $k_1 = m_1 L_1 g, k_2 \neq m_2 L_2 g, P \neq 0$.

We use the computations of the case 2 and describe a locally closed set via the following system of equations and inequations

$$k_1 = m_1 L_1 g, k_2 - m_2 L_2 g \neq 0, k_2 - m_2 L_2 g \neq \frac{m_1 L_1^2 d_2 - m_2 L_2^2 d_1}{m_1^2 L_1^4} d_1$$

In order to treat both inequations involving $k_2 - m_2 L_2 g$, we introduce a new parameter u (thus, u is mutually non-zero in the ground field) and plug in the transposed system module the fake equation $k_2 - m_2 L_2 g = u$.

Also this system is generically neither controllable nor autonomous. The torsion submodule is annihilated by $\langle \partial \rangle$. Compare with the annihilator for the case 2, which is $\langle m_1 L_1^2 \partial^2 + d_1 \partial \rangle$. Let us investigate, for which u the properties change.

```
LIB "control.lib";
ring r3 = (0,g,m1,m2,L1,L2,d1,d2,k1,k2,u),(d),(c,dp);
poly z1 = 0; poly z2 = u;
module RR =
    [m1*L1^2*d^2+d1*d+z1, 0, m1*L1*d^2],
    [0, m2*L2^2*d^2+d2*d+z2, m2*L2*d^2];
module R = transpose(RR);
module S = groebner(R);
matrix T = lift(R,S);
genericity(T);
```

The output of `genericity` delivers

```
[1]:
    u,m2,L2,d1
[2]:
    m1^2*L1^4*u-m1*L1^2*d1*d2+m2*L2^2*d1^2
```

That is, the generic annihilator of a torsion submodule of the system subject to constraints $k_1 - m_1 L_1 g = 0, k_2 - m_2 L_2 g = u \neq 0$ is indeed $\langle \partial \rangle$. However, if $u = k_2 - m_2 L_2 g = \frac{m_1 L_1^2 d_2 - m_2 L_2^2 d_1}{m_1^2 L_1^4} d_1$, the non-generic annihilator equals $\langle m_1 L_1^2 \partial^2 + d_1 \partial \rangle$. This illustrates the difference between two components, corresponding to cases 2 and 3.

Case 4). $k_1 \neq m_1 L_1 g, k_2 = m_2 L_2 g, P = 0$ and

Case 5). $k_1 \neq m_1 L_1 g, k_2 = m_2 L_2 g, P \neq 0$.

The simplified presentation matrix for the transposed module is symmetric, that is, exchanging $m_1 \leftrightarrow m_2, L_1 \leftrightarrow L_2, d_1 \leftrightarrow d_2$ and $k_1 \leftrightarrow k_2$ simultaneously does not change the matrix. Hence, we can take the results of case 2 respectively case 3,

exchange the variables and get the results for case 4 respectively case 5.

Case 6). $k_1 \neq m_1 L_1 g, k_2 \neq m_2 L_2 g, P = 0$.

Recall the special structure of a polynomial P in (1). It is easy to see, that if $P = 0$ and one of the two summands of P is zero, so does the other. This observation lead us to the first conclusion:

$$P = 0, \text{ if } \frac{L_2}{L_1} = \frac{z'_2}{z'_1} = \frac{d'_2}{d'_1}.$$

Going back to the original variables, it translates into

$$(2) \quad \frac{m_2 L_2^2}{m_1 L_1^2} = \frac{k_2 - m_2 L_2 g}{k_1 - m_1 L_1 g} = \frac{d_2}{d_1}.$$

This is especially interesting, since the values, found in [12] for showing the non-generic non-controllability, were $m_1 = m_2 = M_0 = 1, L_1 = L_2 = 1, d_1 = d_2 = 1$ and $k_1 = k_2 = k$. As we can see, it suffices to set $m_1 = m_2, d_1 = d_2, k_1 = k_2 \neq m_2 L_2 g$ and $L_1 = L_2$ for illustrating this phenomenon.

Let us denote by a parameter t the value of the fractions in 2. Then,

$$d_2 = t \cdot d_1, k_2 = t \cdot k_1 + (m_2 L_2 - t \cdot m_1 L_1)g, m_2 L_2^2 = t \cdot m_1 L_1^2$$

We do the substitutions for d_2 and k_2 . As a preprocessing before Gröbner bases, we can manipulate the generators. Consider the last generator of a transposed module, that is, the last column of the transposed presentation matrix $(m_1 L_1 \partial^2, m_2 L_2 \partial^2)^T$. By multiplying the column with L_2 , we can simplify it subject to the substitution to the column $(L_2 \partial^2, t L_1 \partial^2)^T$. The second generator of the module becomes then $(0, t \cdot (m_1 L_1^2 \partial^2 + d_1 \partial + z_1))^T$, from which we cancel the parameter t out. With the following code we perform the controllability and the autonomy analysis for this particular case.

```
ring r6 = (0,g,t,m1,L1,L2,d1,k1),(d),(c,dp);
poly z1 = k1 - m1*L1*g;
module RR =
    [m1*L1^2*d^2+d1*d+z1, 0, L2*d^2],
    [0, m1*L1^2*d^2+d1*d+z1, t*L1*d^2];
module R = transpose(RR);
print(R);
list LC = canonize(control(R));
list LA = canonize(autonom(R));
```

We conclude that this system is neither controllable nor autonomous. In particular, the annihilator of the torsion submodule is the ideal $\langle m_1 L_1^2 \partial^2 + d_1 \partial + k_1 - m_1 L_1 g \rangle$. Note that in view of the equation (2), we obtain the equivalent symmetric annihilator $\langle m_2 L_2^2 \partial^2 + d_2 \partial + k_2 - m_2 L_2 g \rangle$ by e.g. multiplying the previous annihilator with the constant t .

Now let us assume, that $P = 0$ but neither of its summands vanishes. The polynomial P is quadratic with respect to any of the variables $m_1, m_2, k_1, k_2, d_1, d_2$ and is quartic with respect to L_1 and L_2 . Let us fix one of the variables $m_1, m_2, k_1, k_2, d_1, d_2$. Consider the rest of variables as parameters and compute the discriminant of a corresponding quadratic equation. Since the involved variables might have only positive real values, we obtain a condition on the discriminant of a quadratic equation.

If we fix m_1, k_1 or d_1 , we get $\frac{d_2^2}{4m_2L_2^2} \geq k_2 - m_2L_2g$. For fixed m_2, k_2 or d_2 , we obtain, either by a direct computation or via the symmetry, that $\frac{d_1^2}{4m_1L_1^2} \geq k_1 - m_1L_1g$. Notably, both inequalities cannot become equalities simultaneously.

Provided $\frac{d_2^2}{4m_2L_2^2} \geq k_2 - m_2L_2g$, the explicit solution with respect to, say, d_1 gives the following expression (recall, we use the short notation $z_i = k_i - m_iL_ig$):

$$d_1 = \frac{d_2(m_2L_2^2z_1 + m_1L_1^2z_2) \pm (m_2L_2^2z_1 - m_1L_1^2z_2)\sqrt{d_2^2 - 4m_2L_2^2z_2}}{2m_2L_2^2z_2}$$

Each root corresponds to a separate system. Substituting the roots into our system, we obtain, that as in all previous cases, it is neither controllable nor autonomous. The annihilators of torsion submodules are then $\langle 2m_2L_2^2\partial + d_2 \pm \sqrt{d_2^2 - 4m_2L_2^2z_2} \rangle$. The annihilators with respect to m_1, d_1, L_1, z_1 we obtain by the symmetry.

Finally, we summarize the obtained results.

Proposition 4.1. The complete stratification of the obstructions to genericity for the generically controllable system with the essential friction is obtained. All the components of the stratification correspond to non-controllable and non-autonomous systems, whose torsion submodules are annihilated by one of the ideals (for $i = 1, 2$)

$$\langle m_iL_i^2\partial^2 + d_i\partial \rangle, \langle m_iL_i^2\partial^2 + d_i\partial + k_i - m_iL_ig \rangle, \langle \partial \rangle,$$

$$\text{and } \langle 2m_iL_i^2\partial + d_i \pm \sqrt{d_i^2 - 4m_iL_i^2(k_i - m_iL_ig)} \rangle, \text{ provided } k_i \leq \frac{d_i^2}{4m_iL_i^2} + m_iL_ig.$$

5. CONCLUSION AND FUTURE WORK

We have investigated the parameter-dependence of structural properties (such as torsion-freeness) of modules over certain algebras over $\mathbb{K}(p_1, \dots, p_t)$, where \mathbb{K} is a ground field and p_i are parameters. The central idea is to keep track of all polynomial expressions in the p_i that occur as denominators during Gröbner basis computation. These problems have practical applications in control theory as outlined in the Introduction. We have shown several nontrivial phenomena that arise with these questions in terms of illustrative worked examples. Our goal for the future is to extend this approach to the study of more general parametric module properties, leading to the implementation of systematic procedures for such problems.

In particular, one is interested in working with parameters, on which the involved operators act nontrivially. That is, the parameters may correspond to $(q-)$ -differentiable and/or $(q-)$ -shiftable functions. Then, the field $\mathbb{K}(p_1, \dots, p_t)$ must be a differential and/or a difference field. The obstructions to genericity are then presented as systems of differential-difference algebraic equations (DDAE) instead of just algebraic equations treated in this article. Though the main principles remain the same, there is a strong need for specialized techniques and systematic computer-algebraic support for both theoretical and implementational parts of the further research in this area. The case of differentiable parameters was treated in the articles [14, 15], the software package OREMODULES [3] seems to be able to provide computational support for this case.

Yet another important direction of investigation is the analysis of numerical phenomena, namely inexact computations with parameters defined as floating point numbers or as certain inequalities. The generalization of our approach to these domains seems to be possible with the help of e.g. cylindrical algebraic decomposition techniques. Alternatively, one may first obtain an exact symbolic solution to parametric problem, say, in form of the complete stratification, and postprocess it with numerical or symbolical–numerical tools.

Acknowledgements. The authors are grateful to A. Quadrat and J.-F. Pommaret for discussions on theoretical methods as well as particular applications. The first author is grateful to the SFB project F1301 of the Austrian FWF for partial financial support.

REFERENCES

- [1] Becker, M., Levandovskyy, V. and Yena, O. A SINGULAR 2.0 library for computations in control theory `control.lib`, 2003. Available from <http://www.singular.uni-kl.de>.
- [2] Chyzak, F., Quadrat, A. and Robertz, D. Effective algorithms for parametrizing linear control systems over Ore algebras. *Applicable Algebra in Engineering, Communications and Computing*, 16(5):319–376, 2005.
- [3] Chyzak, F., Quadrat, A. and Robertz, D. OREMODULES: A symbolic package for the study of multidimensional linear systems. In J. Chiasson and J.-J. Loiseau, editor, *Applications of Time-Delay Systems*, pages 233–264. Springer LNCIS 352, 2007. Available from <http://wwwb.math.rwth-aachen.de/OreModules/>.
- [4] Greuel, G.-M. and Pfister, G. and Schönemann H. SINGULAR 3.0. A Computer Algebra System for Polynomial Computations. Centre for Computer Algebra, University of Kaiserslautern, 2005. Available from <http://www.singular.uni-kl.de>.
- [5] Greuel, G.-M. and Pfister, G. with contributions by Bachmann, O., Lossen, C. and Schönemann, H. *A SINGULAR Introduction to Commutative Algebra*. Springer, 2002.
- [6] Kredel, H. *Solvable polynomial rings*. Shaker, 1993.
- [7] Levandovskyy, V. Non-commutative Computer Algebra for polynomial algebras: Gröbner bases, applications and implementation. *Doctoral Thesis, Universität Kaiserslautern*, 2005. Available from <http://kluedo.ub.uni-kl.de/volltexte/2005/1883/>.
- [8] Levandovskyy, V. and Schönemann, H. Plural — a computer algebra system for noncommutative polynomial algebras. In *Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC'03)*. ACM Press, 2003.
- [9] Levandovskyy, V. and Zerz, E. Computer algebraic methods for the structural analysis of linear control systems. *Proceedings in Applied Mathematics and Mechanics (PAMM)*, 5:717–718, 2005. DOI: 10.1002/pamm.200510333.
- [10] Levandovskyy, V. and Zerz, E. Algebraic systems theory and computer algebraic methods for some classes of linear control systems. In *Proc. of the International Symposium on Mathematical Theory of Networks and Systems (MTNS'06)*, pages 536–541, 2006.
- [11] Leykin, A. Constructibility of the Set of Polynomials with a Fixed Bernstein-Sato Polynomial: an Algorithmic Approach. *Journal of Symbolic Computation*, 32(6):663–675, 2001. DOI: 10.1006/jsc.2001.0488.
- [12] Polderman, J. and Willems, J. *Introduction to mathematical systems theory. A behavioral approach*. Texts in Applied Mathematics. 26. Springer New York, 1997.
- [13] Pommaret, J.-F. *Partial differential control theory*. Mathematics and its Applications 530. Dordrecht: Kluwer Academic Publishers, 2001.
- [14] Pommaret, J.-F. and Quadrat, A. Formal obstructions to the controllability of partial differential control systems. In *Proceedings of the 15th IMACS World Congress, Berlin (Germany)*, volume 5, pages 209–214, 1997.
- [15] Pommaret, J.-F. and Quadrat, A. Formal elimination for multidimensional systems and application to control theory. *Math. Control Signals Syst.*, 13(3):193–215, 2000.
- [16] Tsai, H. and Leykin, A. D-modules package for MACAULAY 2 – algorithms for D-modules, 2006. Available from <http://www.ima.umn.edu/~leykin/Dmodules/>.

- [17] Walther, U. Cohomology, stratifications and parametric Gröbner bases in characteristic zero. *Journal of Symbolic Computation*, 35(5):527–542, 2003.
- [18] Weispfenning, V. Canonical comprehensive Gröbner bases. *Journal of Symbolic Computation*, 36(3–4):669–683, 2003.

LEHRSTUHL D FÜR MATHEMATIK, RWTH AACHEN, TEMPLERGRABEN 64, D-52062 AACHEN,
GERMANY

E-mail address: [viktor.levandovsky,eva.zerz]@math.rwth-aachen.de